Applied Numerical Mathematics 110 (2016) 110–127

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

A reconstructed central discontinuous Galerkin-finite element method for the fully nonlinear weakly dispersive Green–Naghdi model

Haiyun Dong [a,b], Maojun Li [a,b,*]

[a] College of Mathematics and Statistics, Chongqing University, Chongqing, 401331, PR China
[b] Institute of Computing and Data Sciences, Chongqing University, Chongqing, 400044, PR China

ARTICLE INFO

Article history:
Received 10 September 2015
Received in revised form 26 April 2016
Accepted 22 August 2016
Available online 26 August 2016

Keywords:
Green–Naghdi model
Central discontinuous Galerkin methods
Finite element methods
High order methods
Computational cost

ABSTRACT

In this paper, we present a class of high order reconstructed central discontinuous Galerkin-finite element methods for the fully nonlinear weakly dispersive Green–Naghdi model, which describes a large spectrum of shallow water waves. In the proposed methods, we first reformulate the Green–Naghdi model into conservation laws coupled with an elliptic equation, and then discretize the conservation laws with reconstructed central discontinuous Galerkin methods and the elliptic equation with continuous FE methods. The reconstructed central discontinuous Galerkin methods can be viewed as a class of fast central discontinuous Galerkin methods, in which we replace the standard formula for the numerical solution defined on the dual mesh in the central discontinuous Galerkin method with a projection equation in the L^2 sense. The proposed methods reduce the computational cost of the traditional methods by nearly half but still maintain the formal high order accuracy. We study the L^2 stability and an L^2 *a priori* error estimate for smooth solutions of the reconstructed central discontinuous Galerkin method for linear hyperbolic equation. Numerical tests are presented to illustrate the accuracy and computational efficiency of the proposed method.

1. Introduction

The so-called Green–Naghdi models [18,8] are a class of fully nonlinear weakly dispersive shallow water equations, which are applicable to a wide range of problems involving small- to large-amplitude waves on both shallow and relatively deep water. Therefore, many researchers are interested in the development and simulation of Green–Naghdi models in recent years [1,2,4–6,9,17]. We also developed a central discontinuous Galerkin-finite element (CDG-FE) method for the Green–Naghdi model over flat bottom topography and a well-balanced CDG-FE method for the Green–Naghdi model over variable bottom topographies [14].

The underlying central discontinuous Galerkin (CDG) methods, originally proposed by Liu and his collaborators for hyperbolic conservation laws [15], are a family of high order numerical methods defined on overlapping meshes. These methods can be systematically formulated with any order of (formal) accuracy and do not employ any numerical flux at element interfaces as in discontinuous Galerkin (DG) methods by evolving two sets of numerical solutions. Therefore, the CDG meth-

* Corresponding author at: College of Mathematics and Statistics, Chongqing University, Chongqing, 401331, PR China.
E-mail addresses: cqdxdhy@163.com (H. Dong), limj@cqu.edu.cn (M. Li).

http://dx.doi.org/10.1016/j.apnum.2016.08.008
0168-9274/

ods have been widely applied to various equations, such as diffusion equations [16], Hamilton–Jacobi equations [12], ideal magnetohydrodynamics equations [11,13], Green–Naghdi equations [14] and Camassa–Holm equation [10].

However, the CDG methods and the CDG-FE methods are time-consuming and use more storage space due to evolving two sets of numerical solutions, compared with the DG methods. For reducing the computational cost of the CDG-FE methods, we present a reconstructed CDG-FE (RCDG-FE) method for the Green–Naghdi model over flat bottom topography. In the RCDG-FE method, we first reformulate the Green–Naghdi model into conservation laws coupled with an elliptic equation as in [14], and then discretize the conservation laws with the reconstructed CDG (RCDG) methods and the elliptic equation with the FE methods. In RCDG methods, we reconstruct the approximate solution defined on the dual mesh by a projection equation in the L^2 sense, so we do not need to evolve the numerical solution defined on the dual mesh by the standard formula in the CDG methods. Since the computational cost of the projection equation is far less than that of the standard formula in the CDG methods and we do not need to calculate the numerical solution on the primal mesh for the elliptic equation in the RCDG-FE methods, the RCDG-FE methods reduce the computational cost of the CDG-FE methods by nearly half but still maintain the high order accuracy.

The remainder of the paper is organized as follows. In section 2, we review the mathematical formulation of the Green–Naghdi model over flat bottom topography and its reformulation. Section 3 is devoted to the numerical methods, including the CDG-FE method and the RCDG-FE method. Then, we study the L^2 stability and an L^2 *a priori* error estimate for smooth solutions of the reconstructed central discontinuous Galerkin method for linear hyperbolic equation in section 4. In section 5, a set of numerical experiments are presented to illustrate the accuracy and computational efficiency of the RCDG method. Finally, concluding remarks are given in section 6.

2. Governing equations

We consider the fully nonlinear weakly dispersive shallow water waves by the Green–Naghdi model over flat bottom topography in one-dimensional space [18],

$$\begin{cases} h_t + (hu)_x = 0\,, \\ (hu)_t + \left(hu^2 + \frac{1}{2}gh^2 + \frac{1}{3}h^3(u_x^2 - u_{xt} - uu_{xx})\right)_x = 0\,, \end{cases} \tag{1}$$

where h is the total water depth, u is the vertically averaged horizontal velocity, g is the gravitational constant. The subscripts t and x denote the partial derivatives with respect to the time variable t and the spatial variable x, respectively.

The model (1) can be reformulated as conservation laws [3,9,14]

$$\begin{cases} h_t + (hu)_x = 0\,, \\ (hK)_t + \left(hKu + \frac{1}{2}gh^2 - \frac{2}{3}h^3u_x^2\right)_x = 0 \end{cases} \tag{2}$$

coupled with an elliptic equation

$$-\frac{1}{3}\left(h^3u_x\right)_x + hu = hK\,. \tag{3}$$

We shall design numerical schemes for this form of the Green–Naghdi model in the following section. For ease of presentation, we rewrite (2) as

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U}, u)_x = 0\,, \tag{4}$$

with $\mathbf{U} = (h, hK)^\top$ and

$$\mathbf{F}(\mathbf{U}, u) = \left(hu, hKu + \frac{1}{2}gh^2 - \frac{2}{3}h^3u_x^2\right)^\top$$

being the flux.

3. Numerical schemes

In this section, we develop numerical schemes for the reformulated Green–Naghdi model (3) and (4). Let $\{x_{j-\frac{1}{2}}\}_j$ be a uniform partition of the computational domain $\Omega = [x_{min}, x_{max}]$ with mesh size Δx. With $x_j = \frac{1}{2}(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}})$, $I_j = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$ and $I_{j+\frac{1}{2}} = (x_j, x_{j+1})$, we define two discrete function spaces, associated with overlapping meshes $\{I_j\}_j$ (the primal mesh) and $\{I_{j+\frac{1}{2}}\}_j$ (the dual mesh), to approximate $\mathbf{U}$,

$$\mathcal{V}_h^C = \mathcal{V}_h^{C,k} = \{\mathbf{v} : \mathbf{v}|_{I_j} \in [P^k(I_j)]^2\,, \forall\, j\}\,,$$
$$\mathcal{V}_h^D = \mathcal{V}_h^{D,k} = \{\mathbf{v} : \mathbf{v}|_{I_{j+\frac{1}{2}}} \in [P^k(I_{j+\frac{1}{2}})]^2\,, \forall\, j\}\,,$$

where $P^k(I)$ denotes the space of polynomials in I with degree at most k, and $[P^k(I)]^2 = \{\mathbf{v} = (v_1, v_2)^\top : v_i \in P^k(I), i = 1, 2\}$ is its vector version. To approximate u, we define two continuous finite element spaces

$$\mathcal{W}_h^C = \mathcal{W}_h^{C,k} = \{w : w|_{I_j} \in P^k(I_j)\ , \forall\, j \text{ and } w \text{ is continuous}\}\ ,$$
$$\mathcal{W}_h^D = \mathcal{W}_h^{D,k} = \{w : w|_{I_{j+\frac{1}{2}}} \in P^k(I_{j+\frac{1}{2}})\ , \forall\, j \text{ and } w \text{ is continuous}\}\ .$$

For simplicity, we present the numerical schemes in the case of the forward Euler method for time discretization. High order time discretizations will be discussed afterward. The numerical schemes evolve two copies of numerical solutions, which are assumed to be available at $t = t_n$, denoted by $\mathbf{U}_h^{n,\star} = (h_h^{n,\star}, (hK)_h^{n,\star})^\top \in \mathcal{V}_h^\star$ and $u_h^{n,\star} \in \mathcal{W}_h^\star$, and we want to find the solutions at $t = t_{n+1} = t_n + \Delta t_n$ with Δt_n being the actual time step in the simulation. Hereinafter, the symbol $\star$ denotes C or D. For the initialization, we project initial functions into $P^k(I_j)$ on I_j (resp. into $P^k(I_{j+\frac{1}{2}})$ on $I_{j+\frac{1}{2}}$) in the L^2 sense, and obtain the approximations $\mathbf{U}_h^{0,C}$ and $u_h^{0,C}$ (resp. $\mathbf{U}_h^{0,D}$ and $u_h^{0,D}$) throughout the domain Ω.

3.1. Central discontinuous Galerkin-finite element methods

In this subsection, we review the CDG-FE method presented in [14] for the reformulated Green–Naghdi model (3) and (4). To get $\mathbf{U}_h^{n+1,\star} = (h_h^{n+1,\star}, (hK)_h^{n+1,\star})^\top$, we look for $\mathbf{U}_h^{n+1,\star} \in \mathcal{V}_h^\star$ such that for any $\mathbf{V}^\star \in \mathcal{V}_h^\star$ with any j,

$$\begin{aligned}\int_{I_j} \mathbf{U}_h^{n+1,C} \cdot \mathbf{V}^C dx = &\int_{I_j} \left(\theta_n \mathbf{U}_h^{n,D} + (1-\theta_n)\mathbf{U}_h^{n,C}\right) \cdot \mathbf{V}^C dx \\ &+ \Delta t_n \int_{I_j} \mathbf{F}(\mathbf{U}_h^{n,D}, u_h^{n,D}) \cdot \mathbf{V}_x^C dx \\ &- \Delta t_n \Big[\mathbf{F}(\mathbf{U}_h^{n,D}(x_{j+\frac{1}{2}}), u_h^{n,D}(x_{j+\frac{1}{2}})) \cdot \mathbf{V}^C(x_{j+\frac{1}{2}}) \\ &- \mathbf{F}(\mathbf{U}_h^{n,D}(x_{j-\frac{1}{2}}), u_h^{n,D}(x_{j-\frac{1}{2}})) \cdot \mathbf{V}^C(x_{j-\frac{1}{2}})\Big]\ . \end{aligned} \tag{5}$$

$$\begin{aligned}\int_{I_{j+\frac{1}{2}}} \mathbf{U}_h^{n+1,D} \cdot \mathbf{V}^D dx = &\int_{I_{j+\frac{1}{2}}} \left(\theta_n \mathbf{U}_h^{n,C} + (1-\theta_n)\mathbf{U}_h^{n,D}\right) \cdot \mathbf{V}^D dx \\ &+ \Delta t_n \int_{I_{j+\frac{1}{2}}} \mathbf{F}(\mathbf{U}_h^{n,C}, u_h^{n,C}) \cdot \mathbf{V}_x^D dx \\ &- \Delta t_n \Big[\mathbf{F}(\mathbf{U}_h^{n,C}(x_{j+1}), u_h^{n,C}(x_{j+1})) \cdot \mathbf{V}^D(x_{j+1}) \\ &- \mathbf{F}(\mathbf{U}_h^{n,C}(x_j), u_h^{n,C}(x_j)) \cdot \mathbf{V}^D(x_j)\Big]\ . \end{aligned} \tag{6}$$

Here $\theta_n = \Delta t_n / \tau_{max} \in [0, 1]$ with τ_{max} being the maximal time step allowed by the CFL restriction (see [15]).

Once $\mathbf{U}_h^{n+1,\star}$ is available, we can obtain $u_h^{n+1,\star}$ by a continuous finite element method: look for $u_h^{n+1,\star} \in \tilde{\mathcal{W}}_h^\star$ such that for any $v^\star \in \hat{\mathcal{W}}_h^\star$,

$$\frac{1}{3}\int_\Omega (h_h^{n+1,C})^3 (u_h^{n+1,C})_x v_x^C dx + \int_\Omega h_h^{n+1,C} u_h^{n+1,C} v^C dx = \int_\Omega (hK)_h^{n+1,C} v^C dx\ . \tag{7}$$

$$\frac{1}{3}\int_\Omega (h_h^{n+1,D})^3 (u_h^{n+1,D})_x v_x^D dx + \int_\Omega h_h^{n+1,D} u_h^{n+1,D} v^D dx = \int_\Omega (hK)_h^{n+1,D} v^D dx\ . \tag{8}$$

Here $\tilde{\mathcal{W}}_h^\star$ and $\hat{\mathcal{W}}_h^\star$ are some variants of $\mathcal{W}_h^\star$ with consideration of the boundary conditions. For instance, when the boundary condition is of Neumann type, $\tilde{\mathcal{W}}_h^\star = \hat{\mathcal{W}}_h^\star = \mathcal{W}_h^\star$ is taken (see [14] for other boundary conditions). We refer to (5)–(8) as the CDG-FE method.

3.2. Reconstructed central discontinuous Galerkin-finite element methods

In order to reduce the computational cost of the CDG-FE method (5)–(8), we present a RCDG-FE method in this subsection which is as follows: look for $\mathbf{U}_h^{n+1,\star} \in \mathcal{V}_h^\star$ such that for any $\mathbf{V}^\star \in \mathcal{V}_h^\star$ with any j,

$$\int_{I_j} \mathbf{U}_h^{n+1,C} \cdot \mathbf{V}^C dx = \int_{I_j} \left(\theta_n \mathbf{U}_h^{n,D} + (1-\theta_n)\mathbf{U}_h^{n,C}\right) \cdot \mathbf{V}^C dx + \Delta t_n \int_{I_j} \mathbf{F}(\mathbf{U}_h^{n,D}, u_h^{n,D}) \cdot \mathbf{V}_x^C dx - \Delta t_n \left[\mathbf{F}(\mathbf{U}_h^{n,D}(x_{j+\frac{1}{2}}), u_h^{n,D}(x_{j+\frac{1}{2}})) \cdot \mathbf{V}^C(x_{j+\frac{1}{2}}) - \mathbf{F}(\mathbf{U}_h^{n,D}(x_{j-\frac{1}{2}}), u_h^{n,D}(x_{j-\frac{1}{2}})) \cdot \mathbf{V}^C(x_{j-\frac{1}{2}})\right]. \tag{9}$$

$$\int_{I_{j+\frac{1}{2}}} \mathbf{U}_h^{n+1,D} \cdot \mathbf{V}^D dx = \int_{I_{j+\frac{1}{2}}} \mathbf{U}_h^{n+1,C} \cdot \mathbf{V}^D dx \tag{10}$$

and $u_h^{n+1,D} \in \tilde{\mathcal{W}}_h^D$ such that for any $v^D \in \hat{\mathcal{W}}_h^D$,

$$\frac{1}{3}\int_\Omega (h_h^{n+1,D})^3 (u_h^{n+1,D})_x v_x^D dx + \int_\Omega h_h^{n+1,D} u_h^{n+1,D} v^D dx = \int_\Omega (hK)_h^{n+1,D} v^D dx\,. \tag{11}$$

We refer to (9)–(11) as the RCDG-FE method. There are two differences between the CDG-method and the RCDG-FE method, one is replacing (6) with (10) which is a projection equation in the L^2 sense, the other is removing the finite element equation (8) due to the fact that the RCDG-FE method does not need the numerical solution u_h^C. These differences make the computational cost of the RCDG-FE method lower than that of the CDG-FE method.

3.3. High order time discretizations

In order to match with high order accuracy in space, strong stability preserving (SSP) high order time discretizations will be used in the numerical tests. In this paper, we use the second order Runge–Kutta method [7]

$$\begin{aligned} U^{(1)} &= U^n + \Delta t_n \mathcal{L}(U^n) \\ U^{n+1} &= \frac{1}{2}U^n + \frac{1}{2}(U^{(1)} + \Delta t_n \mathcal{L}(U^{(1)})) \end{aligned} \tag{12}$$

for time discretization when $k=1$ in the discrete function spaces, and the third order Runge–Kutta method [7]

$$\begin{aligned} U^{(1)} &= U^n + \Delta t_n \mathcal{L}(U^n) \\ U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}(U^{(1)} + \Delta t_n \mathcal{L}(U^{(1)})) \\ U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}(U^{(2)} + \Delta t_n \mathcal{L}(U^{(2)})) \end{aligned} \tag{13}$$

for time discretization when $k=2$ in the discrete function spaces. Here the $\mathcal{L}(U)$ denotes the spatial operator.

4. Analysis of the RCDG for linear hyperbolic equation

In this section, we study the L^2 stability and provide an L^2 *a priori* error estimate for smooth solutions of the RCDG method for linear hyperbolic equation. We consider the following linear hyperbolic equation in one dimensional space

$$U_t + U_x = 0,\ (x,t) \in [a,b] \times [0,T] \tag{14}$$

with periodic or compactly supported boundary conditions.

We define two discrete function spaces, associated with overlapping meshes $\{I_j\}_j$ (the primal mesh) and $\{I_{j+\frac{1}{2}}\}_j$ (the dual mesh),

$$\begin{aligned} V_h^C &= V_h^{C,k} = \{v : v|_{I_j} \in P^k(I_j)\ ,\ \forall\, j\}\ , \\ V_h^D &= V_h^{D,k} = \{v : v|_{I_{j+\frac{1}{2}}} \in P^k(I_{j+\frac{1}{2}})\ ,\ \forall\, j\}\ . \end{aligned}$$

The RCDG method for (14), with forward Euler method for time discretization, is given by: Looking for $U_h^{n+1,\star} \in V_h^\star$, such that for any $v^\star \in V_h^\star$ with any j,

$$\int_{I_j} U_h^{n+1,C} v^C dx = \int_{I_j} \left(\theta_n U_h^{n,D} + (1-\theta_n) U_h^{n,C}\right) v^C dx + \Delta t_n \int_{I_j} U_h^{n,D} v_x^C dx - \Delta t_n \left[U_h^{n,D}(x_{j+\frac{1}{2}}) v^C(x_{j+\frac{1}{2}}) - U_h^{n,D}(x_{j-\frac{1}{2}}) v^C(x_{j-\frac{1}{2}}) \right], \tag{15}$$

$$\int_{I_{j-\frac{1}{2}}} U_h^{n+1,D} \cdot v^D dx = \int_{I_{j-\frac{1}{2}}} U_h^{n+1,C} \cdot v^D dx. \tag{16}$$

Again $\theta_n = \Delta t_n/\tau_m \in [0,1]$ with τ_{max} being the maximal time step allowed by the CFL restriction, that is, $\tau_{max} = C_{cfl}\Delta x$ with a given constant C_{cfl} dictated by stability (see [15]).

The semi-discrete version of the RCDG method can be obtained by taking the limit $\Delta t_n \to 0$ in (15)–(16), which is given by: Find $U_h^\star(\cdot,t) \in V_h^\star$ such that for any $v^\star \in V_h^\star$ with any j

$$\int_{I_j} \left(\frac{\partial}{\partial t} U_h^C\right) v^C dx = \frac{1}{\tau_{max}} \int_{I_j} \left(U_h^D - U_h^C\right) v^C dx + \int_{I_j} U_h^D v_x^C dx - U_h^D(x_{j+\frac{1}{2}},t) v^C(x_{j+\frac{1}{2}}) + U_h^D(x_{j-\frac{1}{2}},t)) v^C(x_{j-\frac{1}{2}}), \tag{17}$$

$$\int_{I_{j-\frac{1}{2}}} U_h^D v^D dx = \int_{I_{j-\frac{1}{2}}} U_h^C v^D dx. \tag{18}$$

4.1. L^2 stability

Theorem 1. *The numerical solution U_h^C and U_h^D of the RCDG method in (17)–(18) satisfies the following L^2 stability condition*

$$\frac{1}{2}\frac{d}{dt}\int_a^b (U_h^C)^2 dx = -\frac{1}{\tau_{max}} \int_a^b \left(U_h^D - U_h^C\right)^2 dx + E \le 0 \tag{19}$$

and

$$\int_a^b (U_h^D)^2 dx \le \int_a^b (U_h^C)^2 dx \tag{20}$$

with a small CFL number, E is given by:

$$E = \sum_j \left(U_h^C(x_j,t) - \{U_h^D(x_j,t)\}\right) \lfloor U_h^D(x_j,t) \rfloor. \tag{21}$$

Herein $\{U_h^D(x_j,t)\} = \frac{1}{2}(U_h^D(x_j^+,t) + U_h^D(x_j^-,t))$ and $\lfloor U_h^D(x_j,t)\rfloor = U_h^D(x_j^-,t) - U_h^D(x_j^+,t)$.

Proof. Eq. (18) is equivalent to

$$0 = \frac{1}{\tau_{max}} \int_{I_{j-\frac{1}{2}}} (U_h^C - U_h^D) v^D dx. \tag{22}$$

Taking the test functions $v^C = U_h^C$ in (17) and $v^D = U_h^D$ in (22), respectively, adding them together and then summing up over j, using the boundary condition, one gets

$$\frac{1}{2}\frac{d}{dt}\int_a^b (U_h^C)^2 dx = \frac{1}{\tau_{max}}\int_a^b \left(U_h^D - U_h^C\right)^2 dx + \sum_j \left(\int_{I_j} U_h^D (U_h^C)_x dx - U_h^D(x_{j+\frac{1}{2}}, t) U_h^C(x^-_{j+\frac{1}{2}}, t) + U_h^D(x_{j-\frac{1}{2}}, t) U_h^C(x^+_{j-\frac{1}{2}}, t) \right). \tag{23}$$

Taking $v^D = (U_h^D)_x$ in (18), one has

$$\int_{I_{j-\frac{1}{2}}} U_h^D (U_h^D)_x dx = \int_{I_{j-\frac{1}{2}}} U_h^C (U_h^D)_x dx\,, \tag{24}$$

Then, (23) can be rewritten as

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}\int_a^b (U_h^C)^2 dx &= \frac{1}{\tau_{max}}\int_a^b \left(U_h^D - U_h^C\right)^2 dx \\
&+ \sum_j \left(\int_{I_j} U_h^D (U_h^C)_x dx + \int_{I_{j-\frac{1}{2}}} U_h^C (U_h^D)_x dx - \int_{I_{j-\frac{1}{2}}} U_h^D (U_h^D)_x dx \right. \\
&\left. - U_h^D(x_{j+\frac{1}{2}}, t) U_h^C(x^-_{j+\frac{1}{2}}, t) + U_h^D(x_{j-\frac{1}{2}}, t) U_h^C(x^+_{j-\frac{1}{2}}, t) \right) \\
&= \frac{1}{\tau_{max}}\int_a^b \left(U_h^D - U_h^C\right)^2 dx \\
&+ \sum_j \left(\int_{x_{j-\frac{1}{2}}}^{x_j} (U_h^C U_h^D)_x dx dy + \int_{x_j}^{x_{j+\frac{1}{2}}} (U_h^C U_h^D)_x dx - \int_{I_{j-\frac{1}{2}}} U_h^D (U_h^D)_x dx \right. \\
&\left. - U_h^D(x_{j+\frac{1}{2}}, t) U_h^C(x^-_{j+\frac{1}{2}}, t) + U_h^D(x_{j-\frac{1}{2}}, t) U_h^C(x^+_{j-\frac{1}{2}}, t)) \right) \\
&= \frac{1}{\tau_{max}}\int_a^b \left(U_h^D - U_h^C\right)^2 dx \\
&+ \sum_j \left(\frac{1}{2}((U_h^D(x^+_{j-1}, t))^2 - (U_h^D(x^-_j, t))^2) \right. \\
&\left. + U_h^C(x_j, t) U_h^D(x^-_j, t) - U_h^C(x_j, t) U_h^D(x^+_j, t)) \right) \\
&= \frac{1}{\tau_{max}}\int_a^b \left(U_h^D - U_h^C\right)^2 dx + E
\end{aligned} \tag{25}$$

Here we also use the boundary condition, E is given by:

$$E = \sum_j \left(\frac{1}{2}((U_h^D(x^+_{j-1}, t))^2 - (U_h^D(x^-_j, t))^2) + U_h^C(x_j, t) U_h^D(x^-_j, t) - U_h^C(x_j, t) U_h^D(x^+_j, t) \right),$$

$$= \sum_j \Big(\frac{1}{2}((U_h^D(x_j^+,t))^2 - (U_h^D(x_j^-,t))^2) + U_h^C(x_j,t)U_h^D(x_j^-,t) - U_h^C(x_j,t)U_h^D(x_j^+,t)) \Big),$$

$$= \sum_j \Big(U_h^C(x_j,t) - \{U_h^D(x_j,t)\} \Big) \lfloor U_h^D(x_j,t) \rfloor . \tag{26}$$

Noting that $\tau_{max} = C_{cfl}\Delta x$, so we can always choose a small CFL number C_{cfl} to make sure

$$\frac{1}{2}\frac{d}{dt}\int_a^b (U_h^C)^2 dx = -\frac{1}{\tau_{max}}\int_a^b \Big(U_h^D - U_h^C\Big)^2 dx + E \le 0 \tag{27}$$

Taking $v^D = U_h^D$ in (18), summing up over j, using the boundary condition and Cauchy–Schwarz inequality, we get (20). □

4.2. L^2 error estimate

We first define $Q^\star$ as the standard L^2 projection into $V_h^\star$, namely, for each j,

$$\int_{I_j} (Q^C w(x) - w(x)) v^C dx = 0, \forall v^C \in P^k(I_j) \tag{28}$$

and

$$\int_{I_{j-\frac{1}{2}}} (Q^D w(x) - w(x)) v^D dx = 0, \forall v^D \in P^k(I_{j-\frac{1}{2}}) \tag{29}$$

To provide the L^2 error estimate, we need the following lemmas [15].

Lemma 1. *For a smooth function $w(x)$, there exists a constant $C_0 > 0$, independent of Δx, depending on $w(x)$ and its derivatives, such that*

$$\|Q^C w(x) - w(x)\| + (\Delta x)^{(1/2)} \|Q^C w(x) - w(x)\|_{\Gamma^C} \le C_0 (\Delta x)^{(k+1)} \tag{30}$$

and

$$\|Q^D w(x) - w(x)\| + (\Delta x)^{(1/2)} \|Q^D w(x) - w(x)\|_{\Gamma^D} \le C_0 (\Delta x)^{(k+1)} \tag{31}$$

where Γ^C and Γ^D denote the set of boundary points of all elements I_j and $I_{j-\frac{1}{2}}$ respectively.

Lemma 2. *For a smooth function $v^\star \in V_h^\star$, there exists a constant $C_0 > 0$, independent of Δx and $v^\star$, such that*

$$\|v_x^\star\| \le C_0 (\Delta x)^{-1} \|v^\star\|, \|v_x^\star\|_{\Gamma^\star} \le C_0 (\Delta x)^{-1/2} \|v^\star\| \tag{32}$$

Now we have an L^2 *a priori* error estimate for the RCDG method by using same techniques in [15].

Theorem 2. *The numerical solution U_h^C and U_h^D of the RCDG method in (17)–(18) for (14) with a smooth initial condition $U(x,0) \in H^{k+1}$ satisfies the following L^2 error estimate*

$$\|U - U_h^C\|^2 \le C(\Delta x)^{2k}, \|U - U_h^D\|^2 \le C(\Delta x)^{2k}, \tag{33}$$

where U is the exact solution of (14).

Proof. We first define

$$B_j(U_h^C, U_h^D; v^C, v^D) = \int_{I_j} \left(\frac{\partial}{\partial t} U_h^C\right) v^C dx - \frac{1}{\tau_{max}} \int_{I_j} \left(U_h^D - U_h^C\right) v^C dx$$
$$- \int_{I_j} U_h^D v_x^C dx + U_h^D(x_{j+\frac{1}{2}}, t) v^C(x_{j+\frac{1}{2}})$$
$$- U_h^D(x_{j-\frac{1}{2}}, t)) v^C(x_{j-\frac{1}{2}}) - \frac{1}{\tau_{max}} \int_{I_{j-\frac{1}{2}}} \left(U_h^C - U_h^D\right) v^D dx\,. \tag{34}$$

It is clear from (17)–(18) that one gets

$$B_j(U_h^C, U_h^D; v^C, v^D) = 0 \tag{35}$$

for all j and all $v^\star \in V_h^\star$. It is also apparent that for the exact solution U of (14) one has

$$B_j(U, U; v^C, v^D) = 0 \tag{36}$$

for all j and all $v^\star \in V_h^\star$. Subtracting (35) from (36), one has the error equation

$$B_j(U - U_h^C, U - U_h^D; v^C, v^D) = 0 \tag{37}$$

for all j and all $v^\star \in V_h^\star$. We now define

$$E^C = Q^C U - U_h^C, E^D = Q^D U - U_h^D \tag{38}$$

and

$$e^C = Q^C U - U, e^D = Q^D U - U. \tag{39}$$

Thus, one has

$$U - U_h^C = E^C - e^C, U - U_h^D = E^D - e^D. \tag{40}$$

Plugging (40) into (37) and taking $v^\star = E^\star$ one gets

$$B_j(E^C, E^D; E^C, E^D) = B_j(e^C, e^D; E^C, E^D). \tag{41}$$

For the left hand side of (41), Theorem 1 gives

$$\sum_j B_j(E^C, E^D; E^C, E^D) = \frac{1}{2}\frac{d}{dt}\int_a^b (E^C)^2 dx + \frac{1}{\tau_{max}}\int_a^b \left(E^D - E^C\right)^2 dx - E, \tag{42}$$

where

$$E = \sum_j \left(E^C(x_j, t) - \{E^D(x_j, t)\}\right) \lfloor E^D(x_j, t) \rfloor\,. \tag{43}$$

For E and the right hand side of (41), we can conclude by using the same technique in [15] and Lemmas 1 and 2

$$E \le \int_a^b (E^C)^2 dx + \int_a^b (E^D)^2 dx + C_0(\Delta x)^{2k} \tag{44}$$

and

$$B_j(e^C, e^D; E^C, E^D) \le \int_{I_j} (E^C)^2 dx + \int_{I_j} (E^D)^2 dx + C(\Delta x)^{2k+1}. \tag{45}$$

From (18), one has

$$\int_{I_{j-\frac{1}{2}}} (Q^D U - U_h^D) v^D dx = \int_{I_{j-\frac{1}{2}}} (Q^D U - U_h^C) v^D dx\,, \tag{46}$$

taking $v^D = E^D$ in (46) and summing over j to obtain

$$\begin{aligned}\int_a^b (E^D)^2 dx &= \int_a^b (Q^D U - U + U - U_h^C) E^D dx \\ &= \int_a^b (e^D + E^C - e^C) E^D dx \\ &\le \frac{1}{2}\int_a^b (E^D)^2 dx + \frac{1}{2}\int_a^b (e^D + E^C - e^C)^2 dx \\ &\le \frac{1}{2}\int_a^b (E^D)^2 dx + \frac{3}{4}\int_a^b (e^D)^2 + (E^C)^2 + (e^C)^2 dx\,, \end{aligned} \tag{47}$$

and thus,

$$\begin{aligned}\int_a^b (E^D)^2 dx &\le \frac{3}{2}\int_a^b (e^D)^2 + (E^C)^2 + (e^C)^2 dx \\ &\le \frac{3}{2}\int_a^b (E^C)^2 dx + C_0(\Delta x)^{2k+2}. \end{aligned} \tag{48}$$

Summing up (45) over j and combing with (42), (44) and (48), one gets

$$\frac{1}{2}\frac{d}{dt}\int_a^b (E^C)^2 dx \le C_0 \int_a^b (E^C)^2 dx + C_0(\Delta x)^{2k}. \tag{49}$$

Combining (49) and (30) gives $\|U - U_h^C\| \le C_0(\Delta x)^k$. It is easy to obtain $\|U - U_h^D\| \le C_0(\Delta x)^k$ by using (31), (40), (48) and (49). □

5. Numerical examples

In this section, numerical experiments are presented to demonstrate the performance of the proposed methods. All tests are performed on a personal laptop computer (Intel Core i5-2410M CPU @ 2.30 GHz with 4 GB memory).

All examples were performed with both P^1 and P^2 approximations. For convenience, we only show P^2 results unless stated otherwise and all reported results are from numerical solutions on the primal mesh $\{I_j\}_j$. Since we only calculate the numerical solution u_h^D on the dual mesh for the RCDG-FE method, we reconstruct the numerical solution u_h^C at the final time on the primal mesh by using a projection equation similar to (10) for showing it. In all simulations, the time step is dynamically determined by [14]

$$\frac{C_{\text{cfl}}\Delta x}{\max(|u|)}\,, \tag{50}$$

where the CFL number C_{cfl} is taken as 0.1 for both P^1 and P^2 approximations. We set $\theta_n = 1$ for computational efficiency and non-dimensionalize the equations such that the gravitational constant g is unity.

The Green–Naghdi model has an exact solitary wave solution [18] given by

$$\begin{cases} h(x,t) = h_1 + (h_2 - h_1)\operatorname{sech}^2\left(\frac{x - Dt}{2}\sqrt{\frac{3(h_2 - h_1)}{h_2 h_1^2}}\right) \\ u(x,t) = D\left(1 - \frac{h_1}{h(x,t)}\right)\,, \end{cases} \tag{51}$$

where h_1 is the typical water depth, h_2 corresponds to the solitary wave crest and $D = \sqrt{gh_2}$ is the wave speed. Unless stated otherwise, this expression will be used as initial conditions in our numerical simulations. In particular, it will serve as the reference solution to evaluate numerical errors in our accuracy tests.

Table 1
The L^1, L^2 and L^∞ errors and orders of accuracy of numerical solution at $t = 10.0$ from the DG method for the 1D linear transport equation.

Mesh	L^1 error (Order)	L^2 error (Order)	L^∞ error (Order)
P^1			
100	3.33E−04	3.72E−04	5.89E−04
200	7.76E−05 (2.10)	8.68E−05 (2.10)	1.48E−04 (1.99)
400	1.90E−05 (2.03)	2.13E−05 (2.03)	3.97E−05 (1.90)
800	4.71E−06 (2.01)	5.29E−06 (2.01)	1.03E−05 (1.95)
P^2			
100	8.16E−07	9.74E−07	2.61E−06
200	1.02E−07 (3.01)	1.21E−07 (3.00)	3.27E−07 (2.99)
400	1.27E−08 (3.01)	1.52E−08 (2.99)	4.10E−08 (2.99)
800	1.58E−09 (3.01)	1.90E−09 (3.00)	5.25E−09 (2.96)

Table 2
The L^1, L^2 and L^∞ errors and orders of accuracy of u_h^C at $t = 10.0$ from the CDG method for the 1D linear transport equation.

Mesh	L^1 error (Order)	L^2 error (Order)	L^∞ error (Order)
P^1			
100	3.83E−04	4.31E−04	7.17E−04
200	7.72E−05 (2.31)	8.63E−05 (2.32)	1.40E−04 (2.36)
400	1.80E−05 (2.10)	2.00E−05 (2.11)	3.09E−05 (2.18)
800	4.41E−06 (2.03)	4.90E−06 (2.03)	7.53E−06 (2.03)
P^2			
100	6.39E−07	7.62E−07	1.85E−06
200	7.95E−08 (3.01)	9.50E−08 (3.00)	2.31E−07 (3.00)
400	9.91E−09 (3.00)	1.19E−08 (3.00)	2.90E−08 (3.00)
800	1.23E−09 (3.01)	1.47E−09 (3.01)	3.50E−09 (3.05)

Table 3
The L^1, L^2 and L^∞ errors and orders of accuracy of u_h^C at $t = 10.0$ from the RCDG method for the 1D linear transport equation.

Mesh	L^1 error (Order)	L^2 error (Order)	L^∞ error (Order)
P^1			
100	5.67E−04	6.37E−04	1.03E−03
200	9.32E−05 (2.60)	1.05E−04 (2.60)	1.73E−04 (2.57)
400	1.93E−05 (2.27)	2.15E−05 (2.29)	3.43E−05 (2.34)
800	4.52E−06 (2.09)	5.04E−06 (2.09)	7.65E−06 (2.16)
P^2			
100	7.50E−07	8.84E−07	2.27E−06
200	9.33E−08 (3.01)	1.10E−07 (3.01)	2.83E−07 (3.00)
400	1.16E−08 (3.00)	1.37E−08 (3.00)	3.54E−08 (3.00)
800	1.45E−09 (3.01)	1.71E−09 (3.01)	4.29E−09 (3.04)

5.1. Comparison with the DG method

In order to compare with the DG method, we consider the linear transport equation in one-dimensional space with smooth initial condition

$$U(x, 0) = \sin(2\pi x)$$

and periodic boundary condition.

We present L^1, L^2 and L^∞ errors and orders of accuracy for the numerical solution at $t = 10.0$ from the DG method, the CDG method and the RCDG method in Tables 1, 2 and 3, respectively. In the DG method, we used the Lax–Friedrich flux. We used Gauss–Legendre quadrature with $k + 1$ points to calculate all integrals in the DG, CDG and RCDG methods for P^k approximation.

It can be observed from these tables that like the DG method and the CDG method, the RCDG method is $(k+1)$th order accurate for P^k approximation with $k = 1, 2$ and therefore it is optimal with respect to the approximation properties of the discrete spaces. To compare the computational efficiency, we list the computing time from the DG method, the CDG method and the RCDG method in Table 4. The results show that the computing time of the CDG method is about 2.06 times of the one of the DG method for P^1 approximation and 2.91 times for P^2 approximation, while for the RCDG method, it is 1.11 times for P^1 approximation and 1.53 times for P^2 approximation. Thus the RCDG is faster than the CDG method, but slower than the DG method, this is due to the fact that $U_h^{n,D}$ is discontinuous at the cell center of I_j, thus the numerical

Table 4
The computing time (unit: seconds) from the DG method, the CDG method and the RCDG method for the 1D linear transport equation.

Mesh	100	200	4000	800
P^1				
T_{DG}	1.20	4.21	15.41	59.37
$T_{CDG}(T_{CDG}/T_{DG})$	2.26 (1.88)	8.52 (2.02)	33.20 (2.15)	129.26 (2.18)
$T_{RCDG}(T_{RCDG}/T_{DG})$	1.28 (1.07)	4.60 (1.09)	17.38 (1.13)	67.39 (1.14)
P^2				
T_{DG}	2.22	8.27	31.57	124.36
$T_{CDG}(T_{CDG}/T_{DG})$	6.13 (2.76)	23.91 (2.89)	94.32 (2.99)	374.65 (3.01)
$T_{RCDG}(T_{RCDG}/T_{DG})$	3.29 (1.48)	12.53 (1.52)	49.09 (1.56)	194.36 (1.56)

Table 5
Case 1: the L^2 and L^∞ errors and orders of accuracy of h and u at $t = 100$ from the RCDG-FE method.

Mesh	L^2 error (Order)	L^∞ error (Order)	L^2 error (Order)	L^∞ error (Order)
	h		u	
P^1				
250	4.06E−01	1.92E−01	3.59E−01	1.62E−01
500	1.21E−01 (1.75)	6.07E−02 (1.66)	1.07E−01 (1.75)	4.64E−02 (1.81)
1000	2.89E−02 (2.07)	1.36E−02 (2.16)	2.56E−02 (2.06)	1.18E−02 (1.98)
2000	6.96E−03 (2.05)	3.35E−03 (2.02)	6.17E−03 (2.05)	2.88E−03 (2.03)
4000	1.70E−03 (2.03)	8.27E−04 (2.02)	1.51E−03 (2.03)	7.11E−04 (2.02)
8000	4.18E−04 (2.02)	2.05E−04 (2.01)	3.70E−04 (2.03)	1.76E−04 (2.01)
P^2				
250	7.78E−02	3.67E−02	7.23E−02	2.90E−02
500	1.06E−02 (2.87)	5.48E−03 (2.74)	9.80E−03 (2.88)	3.88E−03 (2.90)
1000	1.41E−03 (2.91)	8.08E−04 (2.76)	1.29E−03 (2.93)	5.28E−04 (2.88)
2000	1.77E−04 (2.99)	1.06E−04 (2.93)	1.61E−04 (3.00)	6.68E−05 (2.98)
4000	2.15E−05 (3.04)	1.29E−05 (3.04)	1.95E−05 (3.05)	8.06E−06 (3.05)
8000	2.51E−06 (3.10)	1.45E−06 (3.16)	2.26E−06 (3.11)	9.27E−07 (3.12)

Table 6
Case 1: the L^2 and L^∞ errors and orders of accuracy of h and u at $t = 100$ from the CDG-FE method.

Mesh	L^2 error (Order)	L^∞ error (Order)	L^2 error (Order)	L^∞ error (Order)
	h		u	
P^1				
250	4.21E−01	2.04E−01	3.74E−01	1.64E−01
500	1.20E−01 (1.81)	5.87E−02 (1.80)	1.06E−01 (1.81)	4.73E−02 (1.79)
1000	2.87E−02 (2.07)	1.35E−02 (2.12)	2.54E−02 (2.06)	1.18E−02 (2.01)
2000	6.95E−03 (2.05)	3.34E−03 (2.01)	6.17E−03 (2.04)	2.88E−03 (2.03)
4000	1.71E−03 (2.02)	8.29E−04 (2.01)	1.52E−03 (2.02)	7.15E−04 (2.01)
8000	4.26E−04 (2.01)	2.06E−04 (2.01)	3.78E−04 (2.01)	1.78E−04 (2.00)
P^2				
250	7.46E−02	3.32E−02	6.93E−02	2.70E−02
500	1.05E−02 (2.82)	4.72E−03 (2.82)	9.77E−03 (2.83)	3.83E−03 (2.82)
1000	1.42E−03 (2.89)	6.63E−04 (2.83)	1.31E−03 (2.90)	5.37E−04 (2.84)
2000	1.82E−04 (2.97)	8.60E−05 (2.95)	1.67E−04 (2.97)	6.96E−05 (2.95)
4000	2.26E−05 (3.01)	1.07E−05 (3.01)	2.07E−05 (3.01)	8.66E−06 (3.01)
8000	2.72E−06 (3.06)	1.28E−06 (3.06)	2.51E−06 (3.05)	1.04E−06 (3.05)

quadrature $\int_{I_j} U_h^{n,D} v_x^C dx$ is divided into two parts, therefore the computing time spend on one cell is double compared with the DG method.

5.2. Accuracy tests

Firstly, we examine the high order accuracy and the computational cost of the RCDG-FE method by varying the mesh size. We shall consider two cases with different solitary waves and boundary conditions.

For the first case, the initial condition is a solitary wave with $h_1 = 1$ and $h_2 = 1.21$ in (51) located at $x = 0$ and propagating in the positive x-direction. The computational domain is $\Omega = [-150, 250]$ and the final time is $t = 100$. Outgoing boundary conditions are used. We present L^2 and L^∞ errors and orders of accuracy for h and u in Table 5. For a comparison, the results from the CDG-FE method are illustrated in Table 6. The computing time of both methods is shown in Table 7.

Table 7
Case 1: the computing times (unit: seconds) from the RCDG-FE method and the CDG-FE method and the ratio between the times.

Mesh	1000	2000	4000	8000
P^1				
$T_{RCDG-FE}$	0.90	3.35	12.65	50.95
T_{CDG-FE}	1.59	6.15	23.99	94.40
$\frac{T_{RCDG-FE}}{T_{CDG-FE}}$	0.57	0.54	0.53	0.54
P^2				
$T_{RCDG-FE}$	4.10	16.19	64.58	259.52
T_{CDG-FE}	8.08	32.32	125.27	497.75
$\frac{T_{RCDG-FE}}{T_{CDG-FE}}$	0.51	0.50	0.52	0.52

Table 8
Case 2: the L^2 and L^∞ errors and orders of accuracy of h and u at $t = 60$ from the RCDG-FE method.

Mesh	L^2 error (Order)	L^∞ error (Order)	L^2 error (Order)	L^∞ error (Order)
	h		u	
P^1				
100	1.86E−00	1.12E−00	1.34E−00	6.90E−01
200	6.64E−01 (1.49)	4.17E−02 (1.43)	4.23E−01 (1.66)	2.27E−01 (1.61)
400	1.14E−01 (2.54)	7.13E−02 (2.55)	7.09E−02 (2.58)	3.71E−02 (2.61)
800	1.77E−02 (2.69)	1.09E−02 (2.71)	1.09E−02 (2.70)	6.03E−03 (2.62)
1600	2.98E−03 (2.57)	1.83E−03 (2.58)	1.83E−03 (2.57)	1.07E−03 (2.49)
P^2				
100	9.31E−02	6.72E−02	5.83E−02	3.32E−02
200	8.15E−03 (3.51)	6.57E−03 (3.35)	5.04E−03 (3.53)	3.01E−03 (3.47)
400	9.07E−04 (3.17)	7.65E−04 (3.10)	5.56E−04 (3.18)	3.49E−04 (3.11)
800	1.10E−04 (3.04)	9.39E−05 (3.03)	6.73E−05 (3.05)	4.27E−05 (3.03)
1600	1.37E−05 (3.01)	1.17E−05 (3.01)	8.36E−06 (3.01)	5.32E−06 (3.00)

Table 9
Case 2: the L^2 and L^∞ errors and orders of accuracy of h and u at $t = 60$ from the CDG-FE method.

Mesh	L^2 error (Order)	L^∞ error (Order)	L^2 error (Order)	L^∞ error (Order)
	h		u	
P^1				
100	1.57E−00	9.54E−01	1.08E−00	5.67E−01
200	4.40E−01 (1.84)	2.72E−01 (1.81)	2.77E−01 (1.97)	1.45E−01 (1.96)
400	7.50E−02 (2.55)	4.60E−02 (2.56)	4.64E−02 (2.58)	2.53E−02 (2.52)
800	1.26E−02 (2.57)	7.65E−03 (2.59)	7.77E−03 (2.58)	4.50E−03 (2.49)
1600	2.34E−03 (2.43)	1.49E−03 (2.36)	1.44E−03 (2.43)	8.79E−04 (2.36)
P^2				
100	7.53E−02	5.42E−02	4.71E−02	2.56E−02
200	7.53E−03 (3.32)	5.78E−03 (3.23)	4.70E−03 (3.33)	2.84E−03 (3.17)
400	8.77E−04 (3.10)	6.83E−04 (3.08)	5.46E−04 (3.11)	3.32E−04 (3.09)
800	1.08E−04 (3.03)	8.40E−05 (3.02)	6.70E−05 (3.03)	4.12E−05 (3.01)
1600	1.34E−05 (3.01)	1.05E−05 (3.01)	8.35E−06 (3.00)	5.15E−06 (3.00)

The results show that like the CDG-FE method, the RCDG-FE method is also $(k+1)$st order accurate for P^k with $k = 1, 2$ and therefore it is optimal with respect to the approximation properties of the discrete spaces. More importantly, the computing time of the RCDG-FE method is only a little more than half of that of the CDG-FE method.

For the second case, the initial solitary wave with $h_1 = 1$ and $h_2 = 2.25$ in (51) is located at $x = 0$ and propagating in the positive x-direction. The computational domain is $\Omega = [-45, 45]$ with periodic boundary conditions and the final time is $t = 60$ at which the solitary wave has traveled one period and thus is back to the initial position. We list the L^2 and L^∞ errors and orders of accuracy for h and u in Tables 8 and 9, and the computing time in Table 10. The results are similar to the case one.

Table 10
Case 2: the computing times (unit: seconds) from the RCDG-FE method and the CDG-FE method and the ratio between the times.

Mesh	200	400	800	1600
P^1				
$T_{RCDG-FE}$	1.26	10.41	96.86	907.83
T_{CDG-FE}	2.45	20.80	190.67	1745.03
$\frac{T_{RCDG-FE}}{T_{CDG-FE}}$	0.51	0.50	0.51	0.52
P^2				
$T_{RCDG-FE}$	6.65	70.39	549.81	4890.44
T_{CDG-FE}	13.10	139.56	1095.03	9589.10
$\frac{T_{RCDG-FE}}{T_{CDG-FE}}$	0.51	0.50	0.50	0.51

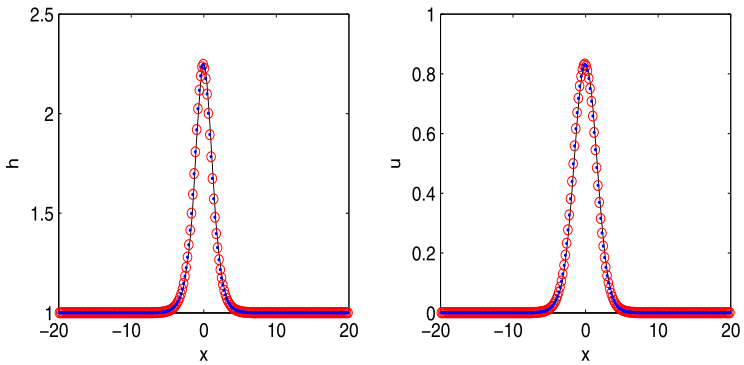

Fig. 1. Solitary wave profiles at $t = 0$ (Solid line) and $t = 900$ (Symbols). Circles: results from the RCDG-FE method; dots: results from the CDG-FE method.

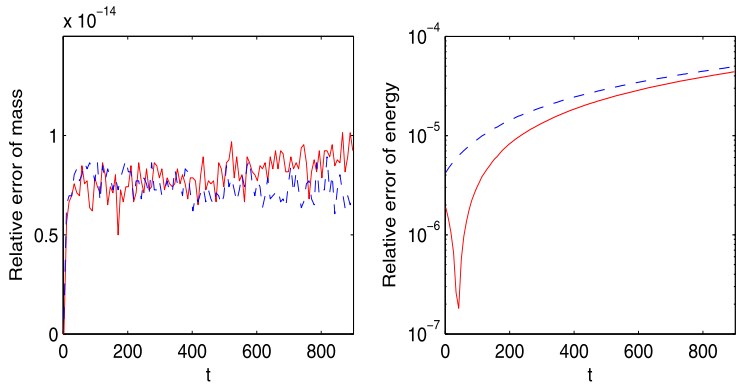

Fig. 2. Time evolution of relative errors on mass and energy for the long-time propagation of a solitary wave. Solid line: results from the RCDG-FE method; dashed line: results from the CDG-FE method.

5.3. Long-time propagation of solitary waves

In this test, we study the performance of the RCDG-FE method when simulating the long-time propagation of a single solitary wave with $h_1 = 1$ and $h_2 = 2.25$, and the computational domain $[-45, 45]$ is discretized with 500 elements. Periodic boundary conditions are used. We compute the solution up to $t = 900$ at which this soliton has traveled 15 lengths of the domain.

The wave profiles from the RCDG-FE method at $t = 900$ are presented in Fig. 1. For a comparison, the profiles at $t = 0$ and the ones at $t = 900$ from the CDG-FE method are also plotted in the same figure. It can be seen from this figure that both methods give similar results and the wave shape and height are well preserved during this long-time propagation compared with the ones at $t = 0$. For a more quantitative assessment, we depict the time evolution of the relative errors on total mass and energy in Fig. 2. These errors are computed relative to the values at $t = 0$. The total mass and energy are

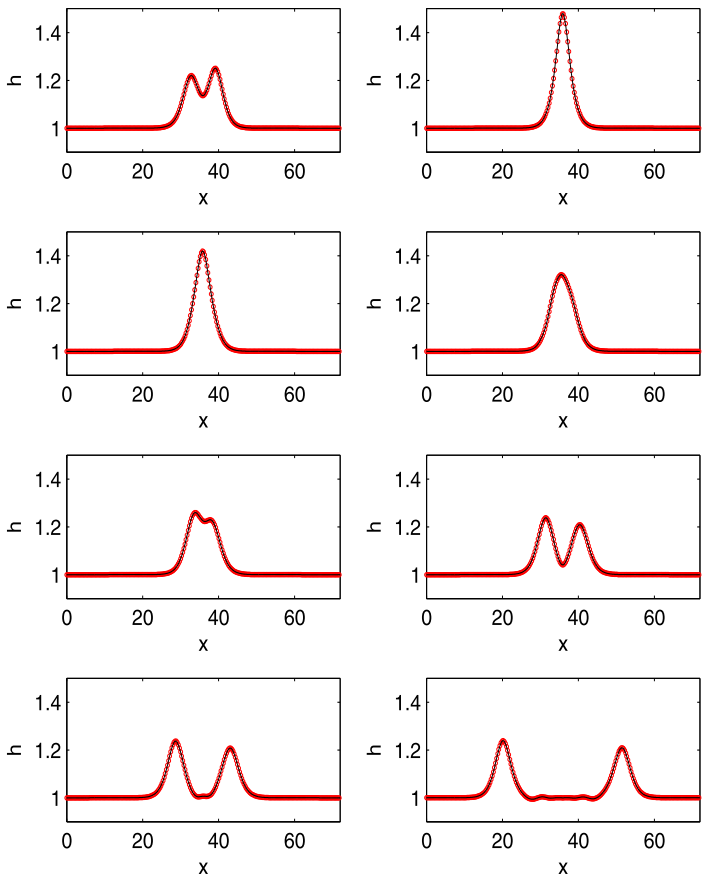

Fig. 3. Water depth (h) for the head-on collision at $t = 7$, 10.5, 11.2, 11.9, 12.5, 14.4, 16.8 and 24.4 (from left to right, from top to bottom). Circles: results from the RCDG-FE method; solid line: results from the CDG-FE method.

defined by [9]

$$M = \int_{\Omega} h\,dx\,, \qquad E = \int_{\Omega} \left(\frac{1}{2}gh^2 + \frac{1}{6}h^3u_x^2\right)dx\,. \tag{52}$$

Fig. 2 indicates that both mass and energy are overall well conserved throughout the computation. For mass conservation, although the relative error from the RCDG-FE method is a little larger than the one from the CDG-FE method at final time, it is still of $O(10^{-14})$ as the one from the CDG-FE method. For energy conservation, the relative errors from both methods are of $O(10^{-5})$ and tend to slowly deteriorate in time. However, the energy conservation from the RCDG-FE method is a little better than the one from the CDG-FE method. For the computing time, the RCDG-FE method takes 2048.48 s, which is 52% of that the CDG-FE method takes.

5.4. Head-on collision of two solitary waves

In this test, we consider the collision between two solitary waves traveling in opposite directions (head-on collision). The head-on collision consists of two initial solitary waves moving toward each other and merging together, finally separating from each other. As a result of this collision, the amplitudes of the two resulting solitary waves are slightly smaller than the initial amplitudes, their centers are slightly retarded from the trajectories of the incoming centers, and there is a small residual from the inelastic nature of the interaction.

In this test, we use two initial solitary waves of different amplitudes. The left wave is initially located at $x = 25$ with $h_2 = 1.213$ in (51) and propagates to the right, while the right wave is initially located at $x = 47$ with $h_2 = 1.243$ and propagates to the left. The typical water depth is $h_1 = 1$ and the computational domain is $[0, 72]$ discretized into 1000 elements. Outgoing boundary conditions are used in this case.

Snapshots of the head-on collision at $t = 7$, 10.5, 11.2, 11.9, 12.5, 14.4, 16.8 and 24.4 are shown in Figs. 3 and 4. Zooming in the solution at $t = 24.4$ clearly reveals these residual waves in Fig. 5. The numerical results from the RCDG-FE method

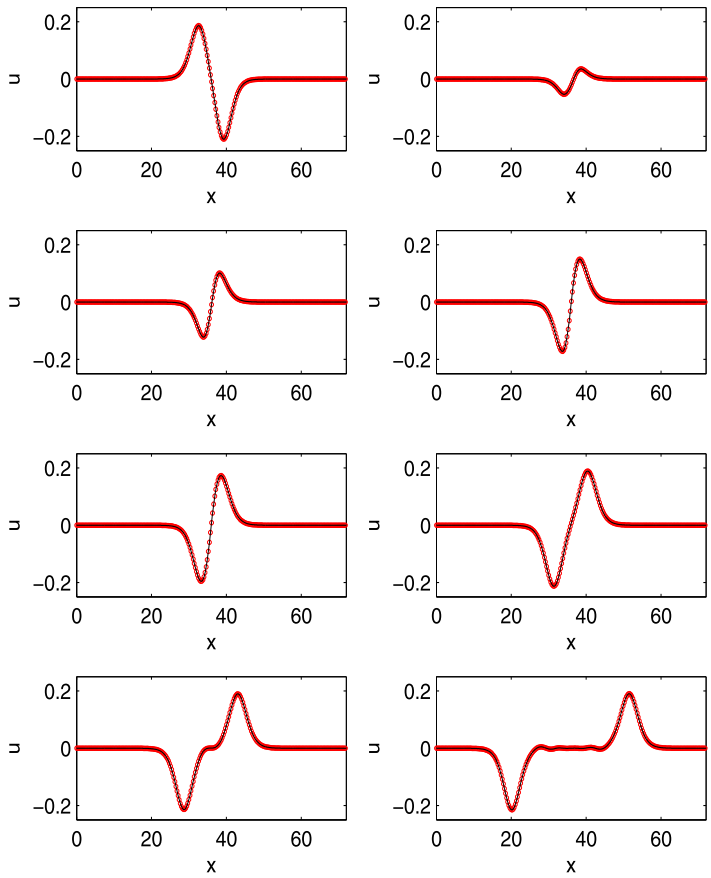

Fig. 4. Water velocity (u) for the head-on collision at $t = 7$, 10.5, 11.2, 11.9, 12.5, 14.4, 16.8 and 24.4 (from left to right, from top to bottom). Circles: results from the RCDG-FE method; solid line: results from the CDG-FE method.

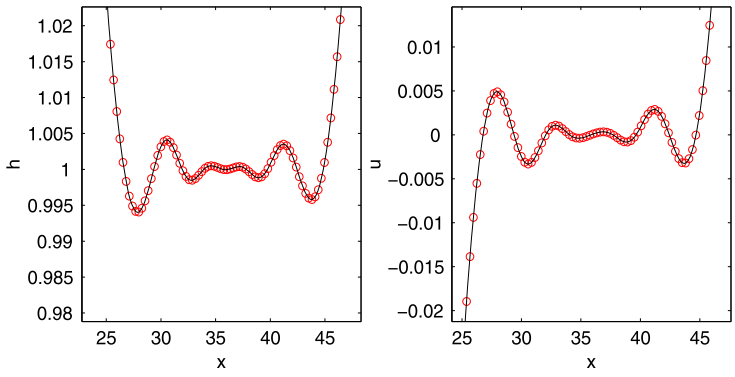

Fig. 5. Zoom-in of residual waves generated by the head-on collision of two solitary waves at $t = 24.4$.

are comparable to the ones from the CDG-FE method. However, the computing time of the RCDG-FE method is only 5.80 s, while it takes 11.28 s for the CDG-FE method.

5.5. Overtaking collision of two solitary waves

In this example, we investigate the collision between two solitary waves traveling in the same direction (overtaking collision). During an overtaking collision, the larger solitary wave catches up and interacts with the smaller one, subsequently passes on and separates from it. The amplitudes and velocities of the solitary waves resulting from an overtaking collision

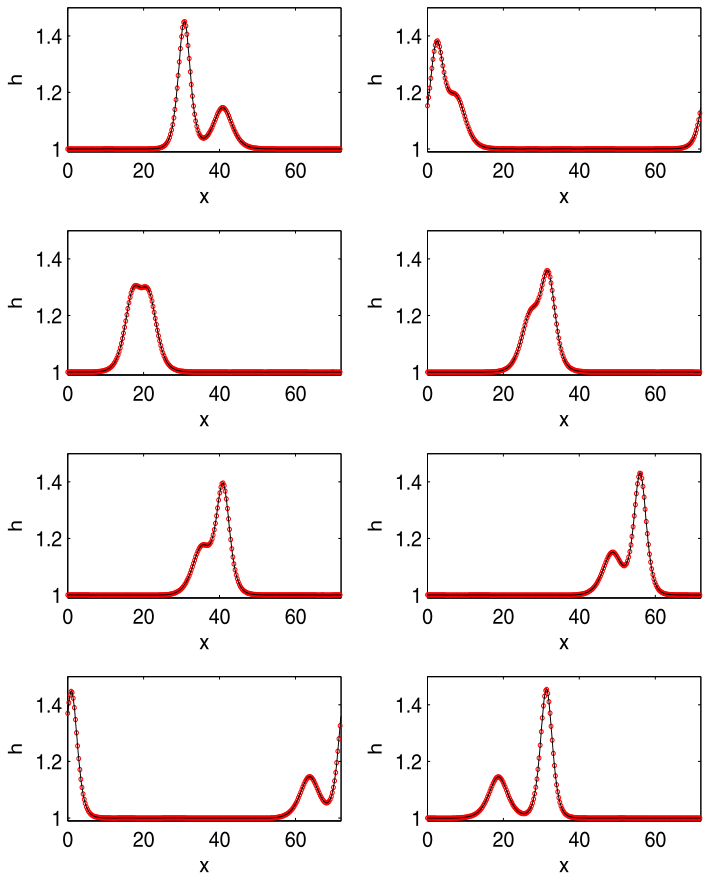

Fig. 6. Water depth (h) for the overtaking collision at $t = 0.0$, 36.40, 49.05, 58.09, 65.79, 78.40, 92.47 and 117.64 (from left to right, from top to bottom). Circles: results from the RCDG-FE method; solid line: results from the CDG-FE method.

have slightly been modified, but the resulting solitary waves experience a substantial positive phase shift. Compared with the case of head-on collisions, the overtaking collision takes place over a long time interval due to the velocities of the same sign of the solitary waves.

In the computation, the computational domain is $[0, 72]$ discretized into 1000 elements. Periodic boundary conditions are used. Snapshots of the overtaking collision at $t = 0.0$, 36.40, 49.05, 58.09, 65.79, 78.40, 92.47 and 117.64 are shown in Figs. 6 and 7. The numerical results from the RCDG-FE method are compared with the results from the CDG-FE method. Both methods give similar results, but it only takes 1891.08 s for the RCDG-FE method to perform this computation, while it takes 3764.46 s for the CDG-FE method to do it.

6. Conclusions

In order to improve the computational efficiency of the CDG-FE method for the Green–Naghdi model, we developed a RCDG-FE method in this paper. The proposed method still maintains the high order accuracy of the CDG-FE method and reduces the computational cost by nearly half. In the future, we will consider the inclusion of the well-balanced and positivity-preserving properties in the RCDG-FE framework for the Green–Naghdi model.

Acknowledgements

Haiyun Dong is partially supported by the National Natural Science Foundation of China (Grant No. 11501062). Maojun Li is partially supported by the Fundamental Research Funds for the Central Universities in China (Project No. 106112015CDJXY100008), the National Natural Science Foundation of China (Grant No. 11501062) and the start-up fund of Chongqing University.

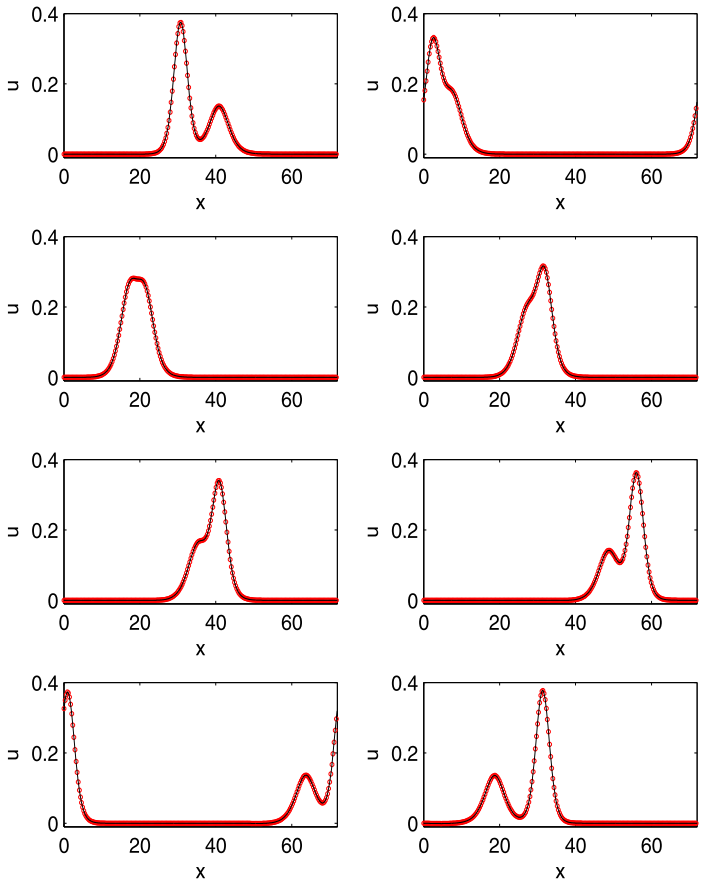

Fig. 7. Water velocity (u) for the overtaking collision at $t = 0.0$, 36.40, 49.05, 58.09, 65.79, 78.40, 92.47 and 117.64 (from left to right, from top to bottom). Circles: results from the RCDG-FE method; solid line: results from the CDG-FE method.

References

[1] P. Bonneton, E. Barthelemy, F. Chazel, R. Cienfuegos, D. Lannes, F. Marche, M. Tissier, Recent advances in Serre–Green Naghdi modelling for wave transformation, breaking and runup processes, Eur. J. Mech. B, Fluids 30 (2011) 589–597.
[2] P. Bonneton, F. Chazel, D. Lannes, F. Marche, M. Tissier, A splitting approach for the fully nonlinear and weakly dispersive Green–Naghdi model, J. Comput. Phys. 230 (2011) 1479–1498.
[3] R. Camassa, D.D. Holm, C.D. Levermore, Long-time effects of bottom topography in shallow water, Physica D 98 (1996) 258–286.
[4] F. Chazel, D. Lannes, F. Marche, Numerical simulation of strongly nonlinear and dispersive waves using a Green–Naghdi model, J. Sci. Comput. 48 (2011) 105–116.
[5] R. Cienfuegos, E. Barthelemy, P. Bonneton, A fourth-order compact finite volume scheme for fully nonlinear and weakly dispersive Boussinesq-type equations. Part I: model development and analysis, Int. J. Numer. Methods Fluids 51 (2006) 1217–1253.
[6] R. Cienfuegos, E. Barthelemy, P. Bonneton, A fourth-order compact finite volume scheme for fully nonlinear and weakly dispersive Boussinesq-type equations. Part II: boundary conditions and validation, Int. J. Numer. Methods Fluids 53 (2007) 1423–1455.
[7] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[8] A.E. Green, P.M. Naghdi, A derivation of equations for wave propagation in water of variable depth, J. Fluid Mech. 78 (1976) 237–246.
[9] O. Le Métayer, S. Gavrilyuk, S. Hank, A numerical scheme for the Green–Naghdi model, J. Comput. Phys. 229 (2010) 2034–2045.
[10] M. Li, A. Chen, High order central discontinuous Galerkin-finite element methods for the Camassa–Holm equation, Appl. Math. Comput. 227 (2014) 237–245.
[11] F. Li, L. Xu, Arbitrary order exactly divergence-free central discontinuous Galerkin methods for ideal MHD equations, J. Comput. Phys. 231 (2012) 2655–2675.
[12] F. Li, S. Yakovlev, A central discontinuous Galerkin method for Hamilton–Jacobi equations, J. Sci. Comput. 45 (2010) 404–428.
[13] F. Li, L. Xu, S. Yakovlev, Central discontinuous Galerkin methods for ideal MHD equations with exactly divergence-free magnetic field, J. Comput. Phys. 230 (2011) 4828–4847.
[14] M. Li, P. Guyenne, F. Li, L. Xu, High order well-balanced CDG-FE methods for shallow water waves by a Green–Naghdi model, J. Comput. Phys. 257 (2014) 169–192.
[15] Y. Liu, C.-W. Shu, E. Tadmor, M. Zhang, Central discontinuous Galerkin methods on overlapping cells with a nonoscillatory hierarchical reconstruction, SIAM J. Numer. Anal. 45 (2007) 2442–2467.
[16] Y. Liu, C.-W. Shu, E. Tadmor, M. Zhang, Central local discontinuous Galerkin methods on overlapping cells for diffusion equations, ESAIM: Math. Model. Numer. Anal. 45 (2011) 1009–1032.

[17] J.D. Pearce, J.G. Esler, A pseudo-spectral algorithm and test cases for the numerical solution of the two-dimensional rotating Green–Naghdi shallow water equations, J. Comput. Phys. 229 (2010) 7594–7608.
[18] C.H. Su, C.S. Gardner, Korteweg-de Vries equation and generalizations. III. Derivation of the Korteweg–de Vries Equation and Burgers Equation, J. Math. Phys. 10 (1969) 536–539.